ERROR GENERATION FOR ADAPTIVE EQUALIZER

BACKGROUND

Equalizers are an important element in many diverse digital information applications, such as voice, data, and video communications. These applications employ a variety of transmission media. Although the various media have differing transmission characteristics, none of them is perfect. That is, every medium induces variation into the transmitted signal, such as frequency-dependent phase and amplitude distortion, multi-path reception, other kinds of ghosting, such as voice echoes, and Rayleigh fading. In addition to channel distortion, virtually every sort of transmission also suffers from noise, such as additive white gausian noise ("AWGN"). Equalizers are therefore used as acoustic echo cancelers (for example in full-duplex speakerphones), video deghosters (for example in digital television or digital cable transmissions), signal conditioners for wireless modems and telephony, and other such applications.

One important source of error is intersymbol interference ("ISI"). ISI occurs when pulsed information, such as an amplitude modulated digital transmission, is transmitted over an analog channel, such as, for example, a phone line or an aerial broadcast. The original signal begins as a reasonable approximation of a discrete time sequence, but the received signal is a continuous time signal. The shape of the impulse train is smeared or spread by the transmission into a differentiable signal whose peaks relate to the amplitudes of the original pulses. This signal is read by digital hardware, which periodically samples the received signal.

Each pulse produces a signal that typically approximates a sinc wave. Those skilled in the art will appreciate that a sinc wave is characterized by a series of peaks

centered about a central peak, with the amplitude of the peaks monotonically decreasing as the distance from the central peak increases. Similarly, the sinc wave has a series of troughs having a monotonically decreasing amplitude with increasing distance from the central peak. Typically, the period of these peaks is on the order of the sampling rate of the receiving hardware. Therefore, the amplitude at one sampling point in the signal is affected not only by the amplitude of a pulse corresponding to that point in the transmitted signal, but by contributions from pulses corresponding to other bits in the transmission stream. In other words, the portion of a signal created to correspond to one symbol in the transmission stream tends to make unwanted contributions to the portion of the received signal corresponding to other symbols in the transmission stream.

This effect can theoretically be eliminated by proper shaping of the pulses, for example by generating pulses that have zero values at regular intervals corresponding to the sampling rate. However, this pulse shaping will be defeated by the channel distortion, which will smear or spread the pulses during transmission. Consequently, another means of error control is necessary. Most digital applications therefore employ equalization in order to filter out ISI and channel distortion.

Generally, two types of equalization are employed to achieve this goal: automatic synthesis and adaptation. In automatic synthesis methods, the equalizer typically compares a received time-domain reference signal to a stored copy of the undistorted training signal. By comparing the two, a time-domain error signal is determined that may be used to calculate the coefficient of an inverse function (filter). The formulation of this inverse function may be accomplished strictly in the time domain, as is done in Zero Forcing Equalization ("ZFE") and Least Mean Square ("LMS") systems. Other methods

involve conversion of the received training signal to a spectral representation. A spectral inverse response can then be calculated to compensate for the channel distortion. This inverse spectrum is then converted back to a time-domain representation so that filter tap weights can be extracted.

In adaptive equalization the equalizer attempts to minimize an error signal based on the difference between the output of the equalizer and the estimate of the transmitted signal, which is generated by a "decision device." In other words, the equalizer filter outputs a sample, and the decision device determines what value was most likely transmitted. The adaptation logic attempts to keep the difference between the two small. The main idea is that the receiver takes advantage of the knowledge of the discrete levels possible in the transmitted pulses. When the decision device quantizes the equalizer output, it is essentially discarding received noise. A crucial distinction between adaptive and automatic synthesis equalization is that adaptive equalization does not require a training signal.

Error control coding generally falls into one of two major categories: convolutional coding and block coding (such as Reed-Solomon and Golay coding). At least one purpose of equalization is to permit the generation of a mathematical "filter" that is the inverse function of the channel distortion, so that the received signal can be converted back to something more closely approximating the transmitted signal. By encoding the data into additional symbols, additional information can be included in the transmitted signal that the decoder can use to improve the accuracy of the interpretation of the received signal. Of course, this additional accuracy is achieved either at the cost of

the additional bandwidth necessary to transmit the additional characters, or of the additional energy necessary to transmit at a higher frequency.

A convolutional encoder comprises a K-stage shift register into which data is clocked. The value K is called the "constraint length" of the code. The shift register is tapped at various points according to the code polynomials chosen. Several tap sets are chosen according to the code rate. The code rate is expressed as a fraction. For example, a ½ rate convolutional encoder produces an output having exactly twice as many symbols as the input. Typically, the set of tapped data is summed modulo-2 (i.e., the XOR operation is applied) to create one of the encoded output symbols. For example, a simple K=3, ½ rate convolutional encoder might form one bit of the output by modulo-2-summing the first and third bits in the 3-stage shift register, and form another bit by modulo-2-summing all three bits.

A convolutional decoder typically works by generating hypotheses about the originally transmitted data, running those hypotheses through a copy of the appropriate convolutional encoder, and comparing the encoded results with the encoded signal (including noise) that was received. The decoder generates a "metric" for each hypothesis it considers. The "metric" is a numerical value corresponding to the degree of confidence the decoder has in the corresponding hypothesis. A decoder can be either serial or parallel—that is, it can pursue either one hypothesis at a time, or several.

One important advantage of convolutional encoding over block encoding is that convolutional decoders can easily use "soft decision" information. "Soft decision" information essentially means producing output that retains information about the metrics, rather than simply selecting one hypothesis as the "correct" answer. For an

overly-simplistic example, if a single symbol is determined by the decoder to have an 80% likelihood of having been a "1" in the transmission signal, and only a 20% chance of having been a "0", a "hard decision" would simply return a value of 1 for that symbol. However, a "soft decision" would return a value of .8, or perhaps some other value corresponding to that distribution of probabilities, in order to permit other hardware downstream to make further decisions based on that degree of confidence.

Block coding, on the other hand, has a greater ability to handle larger data blocks, and a greater ability to handle burst errors.

Figure 1 illustrates a block diagram of a typical digital communication receiver, including channel coding and equalization, indicated generally at 100. The receiver 100 comprises a demodulation and sync component 110, which converts the received analog signal back into a digital format. The receiver 100 further comprises an equalizer 120, an inner decoder 130, a de-interleaver 140, and an outer decoder 150. The inner coding is typically convolutional coding, while the outer coding is typically block coding, most often Reed-Solomon coding. The convolutional and block coding are generally combined in order to exploit the complementary advantages of each.

Figure 2 is a diagram of an equalizer 120 such as is commonly used in the digital receiver 100 shown in Figure 1. Typically, the equalizer 120 includes a controller 228, a finite impulse response ("FIR") filter 222, a decision device 226, and a decision feedback equalizer ("DFE") 224. The FIR filter 222 receives the input signal 221. The FIR filter 222 is used to cancel pre-ghosts—that is, ghost signals that arrive before the main transmission signal. The decision device 226 examines its inputs and makes a decision as to which one of the received signals at its input is the signal to be transmitted to the

output 229. The input to the decision device 226 is modified by a decision feedback equalizer 224, which is used to cancel post-ghosts—that is, ghost signals that arrive after the main transmission signal—and the residual signal generated from the FIR filter.

The decision device 226 is typically a hard decision device, such as a slicer. For example, in an 8VSB system, the slicer can be a decision device based upon the received signal magnitude, with decision values of 0, ±2, ±4, and ±6, in order to sort the input into symbols corresponding to the normalized signal values of ±1, ±3, ±5, and ±7. For another example, the slicer can be multi-dimensional, such as those used in quadrature amplitude modulation ("QAM") systems.

The controller 228 receives the input data and the output data and generates filter coefficients for both the FIR filter 222 and the decision feedback filter 224. Those skilled in the art will appreciate that there are numerous methods suitable for generating these coefficients, including LMS and RLS algorithms.

Figure 3 illustrates further details of the equalizer 120 shown in Figure 2. The input to the decision feedback equalizer 224 is output from the decision device 226, such as a slicer. The sliced data is delayed (F+M) stages, where F equals the number of stages in the FIR filter 222 and M equals the number of stages in the decision feedback equalizer 224. The equalizer 120 then passes the equalized data to a trellis decoder 350. An error signal 310 is generated by subtracting the input to the slicer 226 from its output. The error signal 310 is multiplied by a step size 320 before it is used to update the tap coefficients. Typically, the step size 320 is less than one, in order to permit the error signal to iteratively adjust the tap coefficients over multiple cycles, so that variations in channel response and noise are averaged out. Generally, the smaller the step size, the

more severe the transient conditions under which the equalizer 120 can converge, though at the cost of slower convergence.

Figure 4 shows the further details of a trellis encoder, shown generally at 400, suitable for use with the decision feedback equalizer 224 shown in Figure 3. The trellis encoder 400 is the 8VSB trellis encoder, precoder, and symbol mapper. As will be known by those skilled in the art, the 8VSB trellis encoder 400 uses an 8-level, 3-bit, one dimensional constellation. As can be seen from Figure 4, the 8VSB trellis encoder 400 uses a 2/3 rate trellis code.

Typically, the trellis decoder 350 uses a Viterbi algorithm to decode the signal encoded by the 8VSB trellis encoder 400. Typically, the trellis decoder 350 has a large number of stages—most often 16 or 24. The decoded output 229 is deinterleaved by the de-interleaver 140, and then sent to the outer decoder 150.

Figure 5 shows a typical trellis diagram for an 8VSB trellis code with n stages, shown generally at 500. The heavier line illustrates a current survive path. At each decoding clock cycle a new symbol is sent to the trellis decoder and the survive path is renewed. It will be appreciated that in a VSB system each sample contains one symbol, while in QAM or offset-QAM systems, each sample contains two symbols—one in the I channel, the other in the Q channel. However, regardless of the sample size, the coding and decoding is always performed symbol by symbol. At each stage a decision is made about which state is the most likely (i.e., which symbol was most likely transmitted), based on the survive path. For example, stage 1 gives the first estimation to the input, and stage 2 gives the second estimation to the input, etc. It will be appreciated that the survive path may change based on the decoding process as each new input symbol is

received, so that the survive path may not be the same (though shifted one symbol) from one input sample time to another.

Figure 6 shows the decoding error rate using a typical trellis decoder with the Viterbi decoding algorithm. As can be seen from the graph, when the system is running below the threshold, and even slightly above it, the error rate is lower after decoding, and the greater the decoding stage, the lower the error rate. The graph also shows that the early decoding stages have a higher gain than the later ones.

Figure 7 shows additional details of the equalizer 120. The error signal 310 is taken simply by subtracting a slicer output from a slicer input, then multiplying by the step size 320. The stepped error signal 320 is then multiplied by the input to the FIR filter 222 and DFE 224, and the results sent to accumulators 710 to update the equalizer taps. This error signal only captures the variation between the input signal and the sliced data level. Whenever the sliced data level does not correspond to the originally transmitted data level, the error signal will incorrectly exclude the difference. For example, if a transmitted value of 3 is received as 4.2, the slicer will read the 4.2 as 5, with an error of –0.8. The correct error in this case is actually +1.2. The FIR filter 222 and DFE 224, which use the error signal to correct for channel distortion such as multipathing, will propagate that error.

Therefore, what is needed is an equalizer having a more accurate error signal. The present invention is directed towards meeting this need, as well as providing other advantages over prior equalizers.

SUMMARY OF THE INVENTION

A first embodiment adaptive equalizer has an error signal that is generated by subtracting the output of a trellis decoder from the input to a trellis decoder.

A second embodiment adaptive equalizer comprises: an FIR filter; a trellis decoder coupled to the FIR filter; a mapper coupled to the trellis decoder; and a decision feedback equalizer coupled to the mapper. The decision feedback equalizer receives the mapped and scaled output of the trellis decoder as input. An error signal is generated by subtracting an output of the trellis decoder from the input to the trellis decoder.

A third embodiment adaptive equalizer consists of: an FIR filter; a trellis decoder coupled to the FIR filter; a mapper coupled to the trellis decoder; and a decision feedback equalizer coupled to the mapper. The decision feedback equalizer receives the mapped and scaled output of the trellis decoder as input. An error signal is generated by subtracting an output of the equalizer from the input to the trellis decoder.

A fourth embodiment adaptive equalizer has an error signal that is generated using only a trellis decoder, a mapper, and a decision feedback equalizer.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a prior art digital receiver.

Figure 2 is a diagram of a prior art equalizer suitable for use in the digital receiver of Figure 1.

Figure 3 is a diagram showing further details of the prior art decision feedback equalizer in Figure 2.

Figure 4 is a diagram of a prior art 8VSB trellis encoder, precoder, and symbol mapper.

Figure 5 is a prior art trellis diagram.

Figure 6 is a graph showing the relationship between error rate and signal-to-noise ratio.

Figure 7 is a diagram of the prior art decision feedback equalizer of Figure 3, showing a trellis diagram having n stages.

Figure 8 is a diagram of a preferred embodiment equalizer according to the present invention.

Figure 9 is an alternative embodiment equalizer in which the decision feedback equalizer uses sliced data for its input.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

For the purposes of promoting an understanding of the principles of the invention, reference will now be made to the embodiment illustrated in the drawings and specific language will be used to describe the same. It will nevertheless be understood that no limitation of the scope of the invention is thereby intended, and alterations and modifications in the illustrated device, and further applications of the principles of the invention as illustrated therein are herein contemplated as would normally occur to one skilled in the art to which the invention relates. In particular, although the invention is discussed in terms of an 8VSB system, it is contemplated that the invention can be used with other types of modulation coding, including, for example, QAM and offset-QAM.

Figure 8 illustrates a preferred embodiment equalizer according to the present invention, indicated generally at 800, employing a decoding structure in which the decision feedback equalizer 850 receives its input from the trellis decoder 350. In certain preferred embodiments, the trellis decoder 350 uses a Viterbi algorithm, as is known in the art. The trellis decoder output 803 is input to the decision feedback equalizer 850 via a mapper 810. The mapper 810 maps and scales the trellis decoder's 350 output 803 back to signal levels. For example, in 8VSB the mapper 810 maps and scales the trellis decoder" 350 output 803 back to normalized signal levels of ±1, ±3, ±5, and ±7. In certain embodiments the trellis decoder 350 has 16 stages and the decision feedback equalizer 850 has M taps. In these embodiments, from the 17th tap up to the Mth tap the decision feedback equalizer 850 has the same structure as a traditional decision feedback equalizer 224, except that the input to this section is the mapped and scaled output from

the 16th stage of the trellis decoder. From the 1st tap to the 16th tap the inputs to the DFE 850 are the mapped and scaled version of the output 803 from the 1st to 16th stages of the trellis decoder 350, respectively. As can be seen from Figure 8, for each input symbol there is a survive path (highlighted as a heavier line in Figure 8). The inputs to the DFE 850 from stage 1 to stage 16 are the mapped and scaled decoded output on the survive path.

In certain other embodiments the trellis decoder 350 has some other number of stages "n," and the DFE 850 has M taps. In these embodiments, the decision feedback equalizer 850 has the same structure from the (n+1)th tap up to the Mth tap, and from the 1st tap to the nth tap the inputs to the DFE 850 are the mapped and scaled output from the trellis decoder 350 from the 1st to the nth stage, respectively.

It will be appreciated that the current survive path can change based on the decoding process with each new input symbol, so the survive path may not be the same (though shifted one symbol) from one sample time to the next. Thus, all the inputs to the DFE 850 can vary from symbol to symbol. This is different from prior art DFEs 224, in which the input to the next stage in the equalizer 224 is the delayed symbol from the previous stage.

The equalizer taps are generated as shown in Figure 8. The raw error signal from the summer 860 is taken by subtracting a delayed version of the input to the trellis decoder 350 from the mapped and scaled version of the output 229 of the trellis decoder 350. The raw error signal is then multiplied by the step size 320. The result is then multiplied by the same input to the trellis decoder that was supplied to the summer 860 (to be subtracted from the mapped and scaled output 229) to generate the corrected error

signal. Note that this input to the trellis decoder must be delayed again by the same number of cycles it takes for the error signal 860 to be multiplied by the step size 320. The result is then sent to accumulators 820 to update the equalizer taps.

It will be appreciated that the error signal in an equalizer according to the present invention is generated with a short delay. If the error signal is generated after 16 decoding stages in a 8VSB system, for example, the delay is 192 symbols, due to the 12 parallel encoders used by the 8VSB system. With a symbol rate of 10.76MHz the delay is about 17.8μs. Compared with a maximum channel distortion varying rate of about 200Hz, or 5ms, the delay in generating the error signal is very short. Thus, the delay in error signal generation will not substantially harm convergence due to the varying channel distortion.

However, the present invention can be used even if the number of parallel encoders and decoders is sufficiently large so that the total delay is long enough to potentially harm the tracking of varying channel distortion. In these situations the error signal can be generated at earlier decoding stages. The earlier stages have higher error rate than the last decoding stage, so this is preferable only when necessary to reduce the delay in error signal generation. However, even the earlier decoding stages have a significant gain. Therefore, the result will still be a substantially improved decoding gain over a system in which, for example, the input to the trellis decoder is a sliced signal.

Those skilled in the art will appreciate that a preferred embodiment equalizer as shown in Figure 8 has advantages over prior art equalizers. The input to the decision feedback equalizer has fewer errors, because it is taken from the mapped and scaled output of the trellis decoder. The lower error rate in the trellis decoder's input makes the

equalizer more stable, and causes it to converge more rapidly. Also, the lower error rate in the trellis decoder's input results in a much lower error rate in its output, resulting in a better equalized signal. Furthermore, the equalizer more effectively eliminates long post-ghosts because there is increasing gain from the trellis decoder from stage to stage. There is significant gain starting from the first trellis decoding stage, so that the decision feedback equalizer is benefited from the start. Also, since the trellis decoded output is more reliable and accurate, the input to the decision feedback equalizer can have fewer bits. This permits a reduction in hardware complexity.

It will further be appreciated that these advantages can be achieved without the use of additional hardware, except for a delay line for generating error signals from the decoder output. A standard trellis decoder employing a standard Viterbi algorithm can be used.

Figure 9 illustrates an alternative embodiment equalizer, shown generally at 900, in which the decision feedback equalizer 224 uses sliced data for its input. The error signal 860 is generated by subtracting a delayed version of the slicer input from the trellis decoder output 229. Preferably the trellis decoder output 229 is mapped and scaled back to corresponding data levels with a mapper 910 before it is used to generate the error signal 860. The slicer input is delayed by a number of cycles equal to the number of cycles the trellis decoder 350 uses to generate the output 229. The error signal 860 is then multiplied by the step size 320 before it is used to update the tap coefficients of the FIR filter 222 and the DFE 224.

While the invention has been illustrated and described in detail in the drawings and foregoing description, the same is to be considered as illustrative and not restrictive

in character, it being understood that only the preferred embodiment has been shown and described and that all changes and modifications that come within the spirit of the invention are desired to be protected.